

**TITLE**

**METHOD OF OPENING/CLOSING A DISK TRAY IN A DISK DRIVE**

**BACKGROUND OF THE INVENTION**

**Field of the Invention:**

5       The present invention relates to the operation of a disk drive and particularly to a method of opening/closing a disk tray in the drive.

**Description of the Related Art:**

10       The ATA/IDE interface has become a de facto industry standard for connection of disk drives in PC's. In the interest of simplicity and cost, the ATA/IDE interface was originally designed to support only a small subset of computer peripherals. The expanding use of multimedia, inexpensive program distribution on CD ROM, and faster and  
15 more powerful systems has created the need for enhancements to ATA. The ATAPI CD-ROM specification SFF-8020 is one of those enhancements and provides a simple and inexpensive CD-ROM interface through a superset of ATA.

20       Software applications involving a disk drive load/unload operation, such as those for CD burning or playback, provide a graphical load/eject button, whereby the disk tray can opened or closed by clicking a mouse rather than pressing the physical load/eject button on the front panel of the disk drive. When the load/eject button is  
25 clicked, the status of the disk tray must be acquired to determine whether the load or unload operation is intended. However, the SFF-8020 specification does not define any packet command for acquisition of the disk tray status.

30       Traditionally, an even and odd flag are raised when the load/eject button has been clicked an even and odd number of

times, respectively. The raised even flag indicates that the disk tray is open while the raised odd flag indicates the disk tray is closed. However, since the flag is used locally in only one application, an incorrect disk tray status indication is possible when another application is performing a load/unload operation or the physical load/eject button on the front panel of the disk drive is pressed, as shown in FIG.1A~1E. The disk tray is initially closed, as shown in FIG.1A. The graphical load/eject button is clicked and the disk tray is opened, as shown in FIG. 1B. When the graphical load/eject button is clicked again, the disk tray is closed, as shown in FIG.1C. The user presses the physical load/eject button on the disk drive to open the disk tray, as shown in FIG.1D. Since the graphical load/eject button is not clicked, the even flag is kept raised and wrongly indicates that the disk tray is closed. Thus, when the graphical button is clicked a third time, the disk drive receives a command to open the already open disk tray, which results in no operation as shown in FIG.1E.

To solve this problem, the new SFF-8090 specification defines several packet commands for acquisition of the disk tray status, such as Mechanism Status and Get Notification. However, software applications using these newly defined commands are not applicable to all disk drives, exceptions include the IsoBuster which has not adopted the SFF-8090 standard.

#### SUMMARY OF THE INVENTION

The object of the present invention is to provide a method of opening/closing the disk tray, including calculating the time elapsed between sending of the

Load/Unload Medium command and receiving the returned value from the disk drive. This elapsed time period, instead of the flag used in the traditional method, indicates the disk tray status. Thus, the method of the present invention  
5 eliminates the problem caused by the local flag and is applicable to any disk drive.

The present invention provides a method of opening/closing a disk tray in a disk drive, comprising the steps of when a first event occurs, sending a load/unload  
10 command to the disk drive, to which the disk drive responds with a second or third event, and resending the load/unload command to the disk drive when the disk drive responds to the previous load/unload command with the second event.

The present invention also provides another method of opening/closing a disk tray in a disk drive, comprising the steps of when a first event occurs, sending a load/unload  
15 command to the disk drive and receiving a value returned by the disk drive for the load/unload command, calculating a time period from the sending of the load/unload command to the receiving of the returned value, and resending the  
20 load/unload command when the calculated time period is longer than a threshold time.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

The present invention will become more fully understood  
25 from the detailed description given hereinbelow and the accompanying drawings, given by way of illustration only and thus not intended to be limitative of the present invention.

FIG. 1A~1E are diagrams showing the traditional opening/closing of the disk tray.

FIG. 2A~2E are diagrams showing the opening/closing of the disk tray according to one embodiment of the invention.

FIG. 3 is a flowchart of a method of opening/closing a disk tray in a disk drive according to one embodiment of the invention.

#### DETAILED DESCRIPTION OF THE INVENTION

FIG. 3 is a flowchart of a method of opening/closing a disk tray in a disk drive according to one embodiment of the invention.

10 In step 31, when a first event occurs, such as clicking the graphical the load/eject button, a command Load/Unload Medium defined in SFF-8020 or SFF-8090 is sent to the disk drive. Accordingly, the disk drive responds with a second or third event to be described in the following.

15 In step 32, a first clock value corresponding to the time when the command Load/Unload Medium being sent is recorded. This can be done by a Windows function: GetTickCount. The returned value for the function GetTickCount is stored into a variable OLD, i.e.,  
20 OLD=GetTickCount().

In step 33, a second clock value corresponding to the time when the returned value for the command Load/Unload Medium being received is recorded. This can be achieved by the Windows function GetTickCount. The returned value for  
25 the function GetTickCount is stored into a variable NEW, i.e., NEW=GetTickCount().

In step 34, a difference between the first and second clock value is calculated and, accordingly the time period from the sending of the load/unload command to the receiving  
30 of the returned value is obtained.

In step 35, the time period obtained in step 34 is compared with a threshold value, such as 0.5 seconds. The comparison result determines whether the disk drive responds with the second or third event. The second event sends back  
5 the returned value from the disk drive after a significantly long time period (i.e., longer than the threshold). The third event sends back the returned value from the disk drive immediately (i.e., after a time period shorter than the threshold).

10 In step 36, the command Load/Unload Medium is resent to the disk drive when the disk drive responds to the previous Load/Unload Medium command with the third event, i.e., the disk drive sends back the returned value immediately.

FIGS. 2A~2E are diagrams showing the opening/closing of  
15 the disk tray according to one embodiment of the invention. The disk tray is initially closed, as shown in FIG.2A. When the graphical load/eject button is clicked, the disk tray moves outwards and the returned value is not sent back until the disk tray stops, as shown in FIG. 2B. Since the  
20 returned value is received after a time period longer than 0.5 second, the command Load/Unload Medium is not resent. As shown in FIG.2C, the graphical load/eject button is clicked again, the disk tray moves inwards, and the returned value is not sent back until the disk tray stops. As shown  
25 in FIG.2D, the user presses the physical load/eject button on the disk drive to open the disk tray. When the user clicks the graphical button at the third time, the disk drive receives the Load/Unload Medium command to open the opened disk tray, which results in no operation. Thus, the  
30 returned value is immediately sent back so that the command

Load/Unload Medium is resent to close the disk tray, as shown in FIG. 2E.

In conclusion, the present invention provides a method of opening/closing the disk tray, including calculating the elapsed time period from the sending of the Load/Unload Medium command to the receiving of the returned value from the disk drive. This elapsed time period, instead of the flag used in the traditional method, indicates the disk tray status. Thus, the method of the present invention eliminates the problem caused by the local flag and is applicable to any kind of disk drive.

The foregoing description of the preferred embodiments of this invention has been presented for purposes of illustration and description. Obvious modifications or variations are possible in light of the above teaching. The embodiments were chosen and described to provide the best illustration of the principles of this invention and its practical application to thereby enable those skilled in the art to utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. All such modifications and variations are within the scope of the present invention as determined by the appended claims when interpreted in accordance with the breadth to which they are fairly, legally, and equitably entitled.